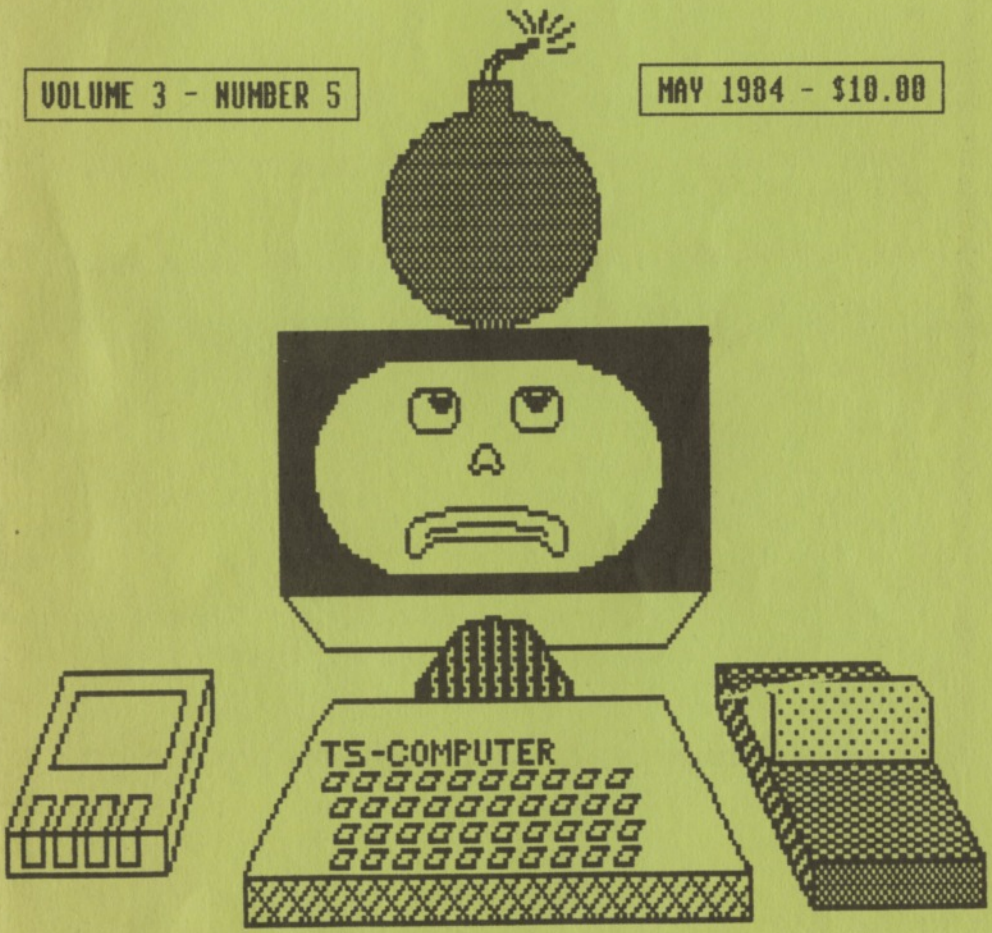
SYNCHRO SETTE

VOLUME 3 - NUMBER 5

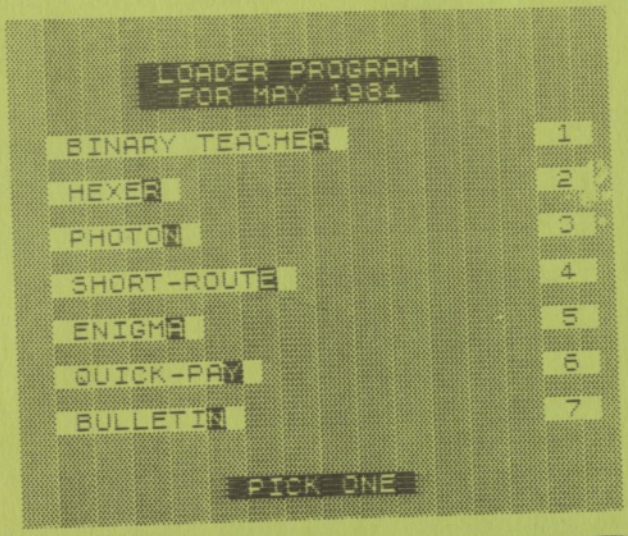
MAY 1984 - $10.00



THE SUBSCRIPTION MAGAZINE FOR
THE T/S-1000 and THE ZX-81 MICROCOMPUTERS

SYNCHRO-SETTE is published monthly by: THE S & S COMPANY
388 W. LAKE ST., ADDISON, IL, 60101 (312)628-8955

4/10U 7-25



This Month's Programs

There are 7 programs on this month's cassette, not counting the LOADER program.

The program names are as they appear on the above menu, which is similar to what will appear on the screen when the LOADER program is running.

Each program is recorded only once on each side of the cassette. The first programs that can be LOADed will be of the small size followed by the larger sized programs. The other side of the cassette is a duplicate of the first side.

For you new subscribers who aren't familiar with LOADing procedures for cassette programs, follow these directions:

A - Make sure that the volume setting of the recorder is set at about 80 % maximum.

B - If you have a Bass and/or Treble control on the recorder, make sure the Treble is at maximum and the Bass is at minimum.

C - To LOAD the first program, type in LOAD "" and press the ENTER key on the computer. Then press the PLAY button of the recorder. The lead time on the NOV/83 cassette is about 10 seconds

Cotc PAGE 3

Vol 4/10 27-47

"BINARY TEACHER" LT = :49

Here is the program that had the listing missing in the last issue. It will easily convert any decimal number to its binary equivalent. More important, an easy to understand screen representation of the binary number will be displayed.

The grid will display in the lower halfof the screen, numbers that are powers of 2. The boxes from right to left represent powers of 2 from 0 to 7. The boxes in the upper half of the grid have either 0s or 1s in them. If a 1 appears, the corresponding power of 2 will be just underneath it. If a 0 appears, no number will appear under it.

The sum of all the lower numbers is the decimal number you entered. The 8 digit number consisting of 0s and 1s is the binary number.

Computers find it easy to use binary numbers because there are only two possibilities in any digit location, a 0 or a 1. This can be thought of as a switch or light being on or off.

Vol 3/4 48-77

"HEXER" LT = 1:14

(Err 2 in 2180)

Here is a game where you match wits against the computer. The computer will scramble the hexidecimal numbers from 0 to F and display the 16 character number under the hexidecimal co-ordinates. You are then asked whether you want to go first. The game will be harder if you let the computer go first.

The idea of the game is for you and the computer to take turns switching hexidecimal numbers from the lower row so that they match the numbers in the upper row. A point will be awarded to the player for each number that matches. A player can get 1, 2 or 0 points each turn.

4/6 Vol 78-91

"PHOTON" LT = :29

Picture a rectangular shaped box with four highly reflective mirror surfaces on the inside. Now picture a single particle of light bouncing off these mirrors and leaving a track of its path that the observer can witness.

This program contains a machine language routine that draws the stationary and moving graphics for this hypothetical situation.

4/10 Vol 92-118

"SHORT-ROUTE" LT = 1:14

One of the most

go to PAGE 6

until the first program begins.

The time needed to load the LOADER program is 48 seconds. When the program is loaded, a list of this month's programs will appear automatically.

Shut off the recorder when the LOADER program is loaded. Any of the listed programs can now be loaded into the computer by pressing the appropriate number on the keyboard and then pressing the PLAY key on the recorder. The loader program loads by searching for the name of the program you want and ignoring any of the other programs it may encounter along the way.

If you want to go directly to a program without waiting, we suggest you first find the tape location of the beginning of each of the programs with your recorder counter. This can be done as you go through the programs the first time, noting the tape location on the counter as each one is being loaded.

If you don't have a counter, approximate the tape position with the fast forward key just before where the program would start, and then LOAD the program with the name of the program using the format LOAD "NAME OF PROGRAM".

Some of our subscribers have told us that they could not get the programs to load by name but they would load with the double quotes. Others have told us that the loader program wouldn't load certain programs. Most have told us that all the programs could be loaded either way. Every customer's cassette is made from the same master tape, so the programs on everyone's cassettes are identical. We feel it is most probably a problem of volume adjustment or recorder design. We have noticed this situation on some of our recorders.

PROGRAMS (all programs this month are self-running - program's name has inverse last character if self-running RT = run time/LT = load time)
There is an approximate 7 to 20 second pause between programs

NOTE! These programs, with minor modification, can be typed into the TS-2068. SCROLL, FAST & SLOW would be eliminated and CODEs and numbers following CHR$ would have to be changed as would any numbers following USR, POKE, PEEK, PLOT or UNPLOT.

time-consuming problems for a computer to overcome is to determine the shortest distance between multiple points on a map. The more points on the map, the more time it takes to solve the problem.

Let us say that you had to visit 10 different cities in your country by driving to each one with your car. Let us say that you wanted to design the shortest route so that you would have to spend the least amount of money on gas. It sounds simple but it isn't. The amount of possibilities to get to all ten cities can be represented by the formula:

10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1

which equals 3,628,800 combinations or possible routes that can be taken. Let us look at it this way! Let us say that we had to visit 3 cities, city A, city B and city C. The combinations would be:

1. A to B to C
2. A to C to B
3. B to A to C
4. B to C to A
5. C to A to B
6. C to B to A

This represents six possibilities or 3 x 2 x 1. Consider the possiblities to visit 20 cities or 100 or more.

Routines can be written into such a search program that will look for short proximity sub-routes to shorten the search time but the bottom line is that the time involved is immense.

A human, however, can usually observe the problem and figure out a reasonable (although not neccessarily perfect) solution by disregarding most of the longer possibilities.

This program is a game that challenges you to move your cursor through a grid (use the arrow keys to start and continue) and remove all the obstacles in a given amount of time. The time given is always proportional to the amount of obstacles and the fewer obstacles there are, the harder it is to complete the trip on time.

Move 119-130

"ENIGMA" LT = :29

This is a program that will demonstrate one of mathematic's oddities. The rules are simple - enter any number between 2 and 8 digits long. In the number, any digit must not repeat itself (sometimes it's OK).

The computer will take

(CONTINUED ON BACK PAGE)

Computer Crash In The Year 2000

THE GREAT COMPUTER CRASH

The year is 2000. We have just celebrated the New Year's holiday. We are back at work on January 2nd behind our computer terminal to enter the day's data entries.

The program prompts us for the date. We enter "010200" which corresponds with the first 2 characters (01) representing January, the third and fourth characters (02) representing the second day of the month and the fifth and sixth characters (00) representing the year!

The computer tries to print out the date on the screen or paper and it comes out "January 2, 1900".

Data files are sorted at the end of the day, week, month or whenever and we find the 2000 dates are at the wrong end of the list.

Dunning letters are sent to customers telling them that accounts are over 100 years past due.

Sound preposterous? This is actually the scenario presented by a book called Computers in Crisis (Petrocelli Books - $32.95) written by Jerome and Marilyn Murray of Glen Ellyn, IL.

At the turn of the century, the banking industry has collapsed. There are no payroll checks, Social Security, pension and welfare checks are lost in electronic limbo, riots occur when credit card conscience people run out of cash and insurance claims go unanswered simply because computers were programmed to handle dates as a 6 character entry.

If you are not a programmer, ask anyone who is how this problem could be overcome with integrated programs (programs that utilize the same data files) and the response will be that it is difficult at best.

Mr. Murray uses Texaco as an example; to convert their existing programs, it would take 91 programmers working full time for 16 years to make the necessary updates.

Mr. Murray suggests

converting the input to an 8 character code where January 2nd of the year 2000 would be represented as "2000.001". The first four characters would represent the year and the rest would be the number of days that have elapsed in that year. A calender routine can be written into the program that will figure out the exact month and day of the month.

Actually, the period in the suggested format can be eliminated for storage purposes so that we wind up with 7 character storage.

But we still have the problem where almost all programs are written to allow only a 6 character entry for the date on a 2 by 2 by 2 basis. Is it possible to put the date into 6 character positions?

It is actually possible to do it in 6!

The answer is to convert the 7 or 8 character format suggested by Mr. Murray into its hexidecimal equivalent.

The year 2000 converted to hexidecimal (using our "BASECON" program) is "7D0". Converting the 2nd day of January to hexidecimal would result in "002". This would give us a date of "7D0002" which of course requires only 6 characters.

This method would allow us to post dates on our invoices up to December 31st of the year 4095. By that time, I'm sure posterity would have computers and be familiar with routines that can handle bases higher than hexidecimal (base 32 would allow dates to be entered long after our sun has turned into a dwarf star).

The hexidecimal method does not, of course, overcome all the problems. Existing date identifying routines scan the 6 characters on a 2 by 2 by 2 decimal basis. New routines would have to be written into all the programs that would operate on a 3 by 3 hexidecimal basis.

I think it would be easier to write these hexidecimal routines into existing programs than to write new decimal routines into the same programs.

In either case, routines would have to be written that would convert the old 6 character format in the new accepted format so that pre-2000 data can be cross-referenced with post-2000 data. On many computers, this data conversion will be most time-consuming - Ed.

Continuation of Programs From 3/84 & 4/84 Issues

CONTINUATION OF PROGRAMS FROM 3/84 AND 4/84

Three programs were left out of the last two issues. Here they are in their entirity.

TS-2068 SCREEN MEMORY MAP
3/84 p. 13

```
 10  GOTO 200
 100 INPUT "Character? ";a$
 110 INPUT "Location? ";c
 120 PRINT AT 0,0;a$
 130 DIM a(8)
 140 FOR n=16384 TO 18176
     STEP 256
 150 LET b=1+(n-16384)/256
 160 LET a(b) = PEEK n
 170 POKE c+n,a(b)
 180 NEXT n
 190 PAUSE 40000
 200 CLS
 210 FOR n=0 TO 21
 220 PRINT AT n,0;" (insert
     30 low dashes ( _ )_
     - SYMBOL/SHIFT 0);n+1
 230 NEXT n
 240 GOTO 100
```

TS-2068 LINE RENUMBERING
4/84 - P.6

```
9985 STOP
9986 INPUT "Starting line
     number? ";s
9987 INPUT "Increment
     number ";i
9988 CLS: LET a=26710:
     LET b=s

9990 POKE a,INT (b/256)
9991 POKE a+1,b-256*INT
     (b/256):PRINT AT 0,0;b
9992 LET b=b+i
9993 LET a=a+i
9994 IF 256*PEEK a+PEEK
     (a+1)= 9985 THEN
     GOTO 9997
9995 IF PEEK (a-1)=13 THEN
     GOTO 9990
9996 GOTO 9993
9997 LIST
9998 INPUT "Do you want to
     delete RENUM? ";y$: IF
     y$(1)="y" THEN DELETE
     9985,9998
```

BINARY TEACHER
4/84 - P.4

This program is on this month's cassette.

The oldest person to pitch in major league baseball was Leroy "Satchell" Paige who was still hurling at age 59.

Editor Ramblings

TIMEX SINCLAIR SHOPPER

If you havn't already received a copy, contact Timex Sinclair Shopper, 67 Elm St., PO Box 250, Camden Maine.

It is a catalog of products for the TS computers from companies that are still in business. The cover states a selling price of $1.95 but I received one free.

They also plan to sell a monthly publication called the TIMEX SINCLAIR NEWS at 12 issues for $29.00.

1,000,000 SPECTRUMS

As of December, 1983, Sinclair has sold over 1 million Spectrum computers outside the U.S. Together with the ZX-80, ZX-81 and TS-1000s, almost 3.5 million computers have been sold.

Timex, on the other hand, has sold about one tenth the amount of TS-2068s as compared to the Spectrum.

BRITISH PUBLICATION

Did you know that in England, Break dancing is called BODY POPPING?

Learn this and many other interesting facts when you subscribe to Popular Computing Weekly, which is touted to be Britain's best selling Micro Weekly publication (I don't know how many competiters it has).

The magazine seems to be dedicated to the Sinclair computers and the Commodore 64 with particular emphasis on the Spectrum. The ZX-81 is still very popular there and there is a lot of software and hardware advertised for the ZX-81 and Spectrum computers.

Subscription (U.S. & outside Europe) prices are 18.70 English pounds for 26 issues or 37.40 English pounds for 52 issues. The current conversion is $1.31 to the pound but check with your bank first. It is best to get a foreign draft for this purpose. The address is:

Popular Computing Weekly
Subscription Dept.
12-13 Little Newport St.
London WC2R 3LD

Letters to the Editor

Dear Ed,

Hi! Do you remember me? I have really missed receiving your Synchro-Sette. One reader put it so well when he stated, "it is like losing an old friend."

Even though we never did meet, I, like many readers, felt like I knew you and would do anything we could do to support such a fine programmer and teacher. With Timex going back to making watches, many of us have been left out in the cold, especially with the TS-2068. Many fine software houses have called it quits. We are lucky to have people like Gary Smith of Hawg Wild Software. He has the Chamelon Eprom and is bringing in programs from England for the Spectrum that will run on the Chamelon/2068. Also E. Arthur Brown has been selling the 2050 modem which I'm happy to report works like a dream on the 2068. They are also bringing in modified hardware from England.

I've been busy lately. I took my first official computer programming class this year in high school. I don't like to brag, but I was far ahead of the class. My teacher noticed this and would let me teach the class a couple times a week. I have you to thank for teaching me to be a good programmer. I'm not a great programmer yet, but I'm well on my way. There have been several contests I've entered and won - the one I'm most proud of is winning first place in the Gaston County Science Expo '84. It was so great beating out a couple of Apples, Commodores, Ataris and my favorite - Trash-80 model 4s. That is what I called the Radio Shack model fours we had in the classrooms. My program was called "CHEMISTRY ASSISTANT" and was designed to be used in the classroom. It is quite long and was originally written to work on the micro-drives that never came

- so I changed it into three 36K programs saved on tape of which the word processor (used to write this letter) was part of one of the programs. It is not the best word-pro in the world, but I like it.

Once again, I want to thank you for all you have done for Timex Sinclair owners. You helped make our computers more than a toy.

C. Day - Gastonia, NC

Dear Charlie,

Your successes are to be applauded. You underestimate your programming prowess. Good luck in your future endeavors. I wish I could be there to see your successes. With the younger computer enthusiasts, the Old Professor seems to have bridged the Generation Gap - Ed.

Dear Ed,

I am amazed at how beautiful and well prepared most programs are - yet, I have heard that some programs may take many hours or even months to be written by an accomplished programmer.

When I observe the listings of some of these programs, I can see that I could type them into the computer in just a few minutes.

Is it really that hard and takes that much time to properly write some of these programs?

R. Peterson - Amherst, WI

Dear Ralph,

One thing that has always amazed me is that the observer only sees the finished program, which for the most part is error free and can command a certain amount of awe and respect from the non-programmer.

Little does this non-programming observer realize that the finished program is usually the result of many stupid mistakes, sometimes numbering in the thousands.

Through the process of trial and error, these mistakes are weeded out and the programs the public sees are the final result. These final versions can usually be improved on also.

Programmers at Atari may work for over a year on a single Game that the public

buys in cartridge form or plays at an Arcade.

Large companies may have business programs that require years of refining to get them to operate at their present state and they may still have bugs in them.

The longest I ever worked on a single program was a little over two years before a workable version was achieved and the first two years, all I did was think about how I wanted it to work. I never entered these thoughts into the computer during that time. The actual programming time was a little over 100 hours and the final version has only 187 lines and takes up 4K of memory. Its description is as follows:

- integrated to accept monthly sales data from files created by customer receipt program.

- accepts only sales amounts and purchase dates from file while ignoring all other data.

- sorts data in chronological order.

- has calendar routine that prints to screen a calender grid for the month of the file. It determines the month and year from the file's name and knows exactly how many days are in the month and which day of the week the first day falls on.

- inserts into each day's box, how many sales were made and the total dollar amount of that day's sales.

- in a box to the right of each week, the week's totals are displayed.

- under each column of days, the sales totals for the days are displayed.

- total month's sales and average sales data are displayed at the top of the screen.

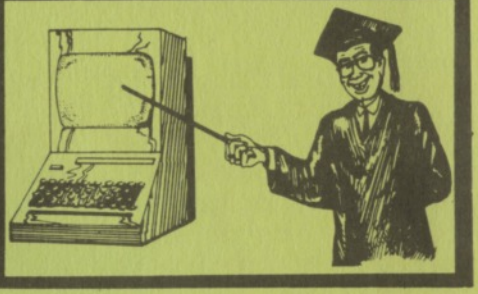
- the entire screen image can be dumped to the printer.

This program is used as an analysis tool where monthly calendar printouts are compared to search for trends. Unlike graph charts that allow analysis on a monthly or at best, weekly basis, the user can see daily performance.

The user can see how holidays, weather, seasons, etc. affect his business. For instance - if for two years in a row, you noticed business picked up for the

 (CONTINUED ON BACK PAGE)

the Computer Tutor



TS-2068 FULL SCREEN USE

Good morning, Class! Many of us were told that we couldn't directly address the bottom 2 lines of the screen on the TS-1000 type computers. A command such as < PRINT AT 22,0;"HELLO" > or < PRINT AT 23,0;"GOODBYE > would result in an error message.

We later found out that if we were to PEEK 16418 (page 135 of the TS-1000 manual), we would find that the number 2 resided in that location. This 2 represented the number of lines at the bottom of the screen unavailable to normal basic programming commands. This number could be changed to a 1 or 0 with a POKE and now the bottom two lines could be accessed with carry-over PRINT commands or PRINT AT commands.

A program incorporating these techniques would work OK until an INPUT command was encountered. The program would then bomb. Re-POKEing a 2 into the location before the INPUT occurred would solve the problem. If you wanted to keep a 0 in the location and only a single character had to be entered from the keyboard, then the INKEY$ function could be used.

On page 263 of the TS-2068 manual, we find the corresponding memory location is 23659 on the TS-2068.

Everything should then operate the same, right? Not so! A program such as the following will cause the TS-2068 to bomb.

```
Version #1
10 POKE 23659,0
20 PRINT AT 22,0;"HELLO"
30 PRINT AT 22,0;"GOODBYE"
40 PAUSE 40000
50 CLS
```

The same thing will happen if program line #40 is changed to <40 GOTO 40> or if it is eliminated. Also if CLS is used after the POKE, the program will bomb.

However, fhe bottom two screen lines can be used if

wanted to keep a 0 in the location and only a single character had to be entered from the keyboard, then the INKEY$ function could be used.

On page 263 of the TS-2068 manual, we find the corresponding memory location is 23659 on the TS-2068.

Everything should then operate the same, right? Not so! A program such as the following will cause the TS-2068 to bomb.

```
Version #1
10 POKE 23659,0
20 PRINT AT 22,0;"HELLO"
30 PRINT AT 22,0;"GOODBYE"
40 PAUSE 40000
50 CLS
```

The same thing will happen if program line #40 is changed to <40 GOTO 40> or if it is eliminated. Also if CLS is used after the POKE, the program will bomb.

However, fhe bottom two screen lines can be used if certain rules are adhered to. The following program will accomplish all the things the above program wouldn't.

```
Version #2
10 POKE 23659,0
20 PRINT AT 22,0;"HELLO"
30 PRINT "GOODBYE"
40 PAUSE 40000
50 POKE 23659,2
60 CLS
```

The reason Version #1 wouldn't work is because program line #30 or #50 will cause the program to bomb. Asking the computer to print at screen line #22 is OK but at line #23, the crash occurs. If only screen line #23 is to be printed at, then program line #20 is changed to read < 20 PRINT AT 22,0 >.

Before the CLS command can be used a < POKE 23659,2 > has to occur.

So, the basic rule of using the bottom two lines of the screen are:

1. POKE location 23659 with a <2>

2. Do not use < PRINT AT 23,0 >

3. Use < PRINT AT 22,0 > to set up screen line #23

4. Do not print at a location beyond the last character position of the 23rd screen line

5. Do not use < CLS > if PEEK 23659 = 0 or 1 (change to 2 first and back to 0 after CLS)

(CONTINUED ON BACK PAGE)

three days before the 4th of July, you may be correct in your decision to stock up on those sales items needed for that holiday.

You may have noticed that business was slow for a week after the 4th for two years in a row. This might make you decide it would be a good idea to take a vacation.

So much for my rambling on (author's prerogative). The bottom line is, I could not find anywhere an outline for this type of program. I had to write it from scratch. The algorithms were unbelievably complicated. I could get it to work properly after about 40 hours but the screen generation was very slow. The whole operation now takes about 45 seconds from the time the file is in the computer until the screen display is finished.

Writing these speed-up routines took the other 60 hours - Ed.

the number, reverse it and then take the difference between those two numbers. It will then take the difference number, reverse it and add the result to the difference number. The result is usually the same for all numbers of the same digit length.

For a 2 digit number the result is 99 and for a 3 digit number it is 1089.

4/10 Vol 131-153

"QUICK-PAY" LT = 1:08

Here is a short payroll program! It won't keep files but it will give a print-out (Z key) after the results are displayed on the screen.

Three formats can be used. The first is for overtime paid at time and a half after an eight hour work day. The second format figures overtime automatically at time and a half for hours after 40. The third is for other schedules such as once or twice a month pay periods.

Vol. 3 1/2/10 Bulletin 154-166

6. Do not use INPUT command if PEEK 23659 = 0 or 1

7. Do not end program while PEEK 23659 = 0 or 1. This will cause program to crash - POKE 23659,2 first.

There may be some other weird effects not covered here so make sure to SAVE the program on cassette periodically while it is being written - Class dismissed.